

CLAIMS

What is claimed is:

1 1. A computer-implemented method for generating edge profile data for executable
2 program code generated from source code, comprising:
3 creating edges that represent respective branch instructions in the executable
4 program code, each edge having associated therewith a source attribute, a target attribute,
5 and an edge-taken count attribute;
6 counting numbers of times the branch instructions are executed and assigning
7 values of the numbers to the edge-taken count attributes of the edges;
8 identifying entry point addresses and target addresses of stub code segments in the
9 executable program code; and
10 for any edge having an edge target attribute that matches an entry point address of
11 a stub code segment, changing the edge target attribute to the stub target address
12 associated with the matching entry point address.

1 2. The method of claim 1, further comprising:
2 patching selected instructions in the executable program with breakpoints;
3 upon encountering an breakpoint during program execution, determining whether
4 instructions that follow the breakpoint comprise a stub code segment; and
5 if a stub code segment is found, associating the entry point address with the target
6 address of the stub code segment.

1 3. The method of claim 2, further comprising:
2 saving the selected instructions patched with breakpoints; and

3 replacing a breakpoint with a corresponding instruction upon encountering the
4 breakpoint.

1 4. The method of claim 3, further comprising:
2 upon encountering an breakpoint during program execution, determining whether
3 instructions that follow the breakpoint comprise a stub code segment or a function; and
4 if a function is found, providing an instrumented version of the function to execute
5 in lieu of the function, wherein the instrumented version of the function increments the
6 values of edge-taken count attributes for edges in the function.

1 5. The method of claim 2, further comprising:
2 upon encountering an breakpoint during program execution, determining whether
3 instructions that follow the breakpoint comprise a stub code segment or a function; and
4 if a function is found, providing an instrumented version of the function to execute
5 in lieu of the function, wherein the instrumented version of the function increments the
6 values of edge-taken count attributes for edges in the function.

1 6. The method of claim 1, further comprising providing as output data values of the
2 edge-taken count attributes in association with source code line numbers associated with
3 the edge source attributes and the edge target attributes.

1 7. An apparatus for generating edge profile data for executable program code
2 generated from source code, comprising:

3 means for creating edges that represent respective branch instructions in the
4 executable program code, each edge having associated therewith a source attribute, a
5 target attribute, and an edge-taken count attribute;

6 means for counting numbers of times the branch instructions are executed and
7 assigning values of the numbers to the edge-taken count attributes of the edges;

8 means for identifying entry point addresses and target addresses of stub code
9 segments in the executable program code;

10 means, for any edge having an edge target attribute that matches an entry point
11 address of a stub code segment, for changing the edge target attribute to the stub target
12 address associated with the matching entry point address.

1 8. A computer-implemented method for generating edge profile data for executable
2 program code generated from source code, comprising:

3 for each branch instruction in the executable program code, storing an address of
4 the branch instruction as an edge source address and storing a target address of the branch
5 instruction as an edge target address;

6 associating source addresses and target addresses of selected ones of branch
7 instructions with source code line numbers;

8 counting numbers of times the branch instructions are executed as respective edge-
9 taken counts;

10 storing entry point addresses of stub code segments in association with target
11 addresses of the stub code segments;

12 for each edge target address that matches an entry point address of a stub code
13 segment, replacing the edge target address with the stub target address associated with the
14 matching entry point address; and

15 associating the edge-taken counts with the edge source addresses.

1 9. The method of claim 8, further comprising providing as output data the edge-taken
2 counts in association with source code line numbers associated with the edge source
3 addresses and the edge target addresses.

1 10. The method of claim 8, further comprising:
2 patching selected instructions in the executable program with breakpoints;
3 upon encountering an breakpoint during program execution, determining whether
4 instructions that follow the breakpoint comprise a stub code segment; and
5 if a stub code segment is found, associating the entry point address with the target
6 address of the stub code segment.

1 11. The method of claim 10, further comprising:
2 saving the selected instructions patched with breakpoints; and
3 replacing a breakpoint with a corresponding instruction upon encountering the
4 breakpoint.

1 12. The method of claim 11, further comprising:
2 upon encountering an breakpoint during program execution, determining whether
3 instructions that follow the breakpoint comprise a stub code segment or a function; and
4 if a function is found, providing an instrumented version of the function to execute
5 in lieu of the function, wherein the instrumented version of the function increments the
6 values of edge-taken count attributes for edges in the function.

1 13. The method of claim 10, further comprising:
2 upon encountering an breakpoint during program execution, determining whether
3 instructions that follow the breakpoint comprise a stub code segment or a function; and
4 if a function is found, providing an instrumented version of the function to execute
5 in lieu of the function, wherein the instrumented version of the function increments the
6 values of edge-taken count attributes for edges in the function.

1 14. An apparatus for generating edge profile data for executable program code
2 generated from source code, comprising:
3 means, for each branch instruction in the executable program code, for storing an
4 address of the branch instruction as an edge source address and storing a target address of
5 the branch instruction as an edge target address;
6 means for associating source addresses and target addresses of selected ones of
7 branch instructions with source code line numbers;
8 means for counting numbers of times the branch instructions are executed as
9 respective edge-taken counts;
10 means for storing entry point addresses of stub code segments in association with
11 target addresses of the stub code segments;
12 means, for each edge target address that matches an entry point address of a stub
13 code segment, for replacing the edge target address with the stub target address associated
14 with the matching entry point address; and
15 means for associating the edge-taken counts with the edge source addresses.